

IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A computer-accessible memory medium configured to that stores program instructions for performing time-bounded execution of a program, wherein the ~~medium is configured~~ program instructions are executable by a processor to perform:

initiating a timed program execution process, wherein the timed program execution process is operable to execute a program in a time-bounded manner;

initiating a timeout process, wherein the timeout process is operable to preempt the execution process to interrupt execution of the program;

configuring a timeout event, wherein the timeout event is an event indicating a timeout condition for the program;

the timed program execution process performing a time-bounded execution of the program, comprising:

a) determining and storing a rollback state for the program;

b) if the timeout event has not occurred, executing the program, wherein, during said executing, if the timeout event occurs,

c) the timeout process setting the timed program execution process to the rollback state, and disabling the timeout event; and

d) the timed program execution process resuming executing the program based on the rollback state with a timeout condition;

e) performing a program exit procedure;

disabling the timeout event;

terminating the timeout process; and

terminating the timed program execution process.

2. (Currently Amended) The memory medium of claim 1, wherein e) further comprises:

clearing the rollback state.

3. (Currently Amended) The memory medium of claim 2, wherein the ~~medium is further configured~~ program instructions are further executable to perform:

iteratively performing said time-bounded execution of a plurality of programs.

4. (Currently Amended) The memory medium of claim 3, wherein said iteratively performing comprises:

for each of the plurality of programs, performing a) through c).

5. (Currently Amended) The memory medium of claim 4,
wherein e) further comprises:

if the timeout event has occurred, storing an indication of a timeout condition; and

wherein the timed program execution process performing a time-bounded execution of the program comprises:

performing a) through c) if the timeout condition is not indicated.

6. (Currently Amended) The memory medium of claim 2, wherein the ~~medium is further configured~~ program instructions are further executable to iteratively perform:

setting a timeout event;

the timed program execution process performing a time-bounded execution of the program; and

disabling the timeout event.

7. (Currently Amended) The memory medium of claim 2, wherein the timeout process executes at a first priority level, and wherein the ~~medium is further configured~~ program instructions are further executable to perform:

setting an execution priority level of the timed program execution process to a second priority level, wherein the second priority level is below the first priority level.

8. (Currently Amended) The memory medium of claim 7,

wherein said performing a time-bounded execution of the program further comprises:

storing an original execution priority level of the timed program execution process prior to said setting the execution priority level of the timed program execution process; and

wherein e) further comprises:

restoring the execution priority level of the timed program execution process to the original execution priority level.

9. (Currently Amended) The memory medium of claim 2,

wherein said initiating the timeout process comprises:

acquiring one or more resources for the time-bounded execution of the program; and

initializing one or more resource managers for the one or more resources;
and

wherein said terminating the timeout process comprises:

releasing the one or more resources for the time-bounded execution of the program; and

un-initializing the one or more resource managers for the one or more resources.

10. (Currently Amended) The memory medium of claim 9,

wherein said initiating the timeout process further comprises:

creating each of the one or more resource managers; and

wherein said terminating the timeout process further comprises:

deleting each of the one or more resource managers.

11. (Currently Amended) The memory medium of claim 9,

wherein said initiating the timeout process further comprises:

enabling each of the one or more resource managers; and

wherein said terminating the timeout process further comprises:

disabling each of the one or more resource managers.

12. (Currently Amended) The memory medium of claim 9, wherein said one or more resources comprises:

one or more memory pools for memory allocations during the time-bounded execution of the program.

13. (Currently Amended) The memory medium of claim 9,

wherein, prior to a), the ~~medium is further configured~~ program instructions are further executable to perform:

storing a respective resource state for each of the one or more resources;

and

enabling each of the one or more resource managers.

wherein e) further comprises:

if the timeout event has occurred,

restoring the respective resource state for each of the one or more resources; and

disabling each of the one or more resource managers.

14. (Currently Amended) The memory medium of claim 1, wherein said initiating the timeout process is performed during said executing.

15. (Currently Amended) The memory medium of claim 1,

wherein the program comprises one or more sub-programs; and

wherein b) further comprises:

performing a) through e) for each of the one or more sub-programs.

16. (Currently Amended) The memory medium of claim 1, wherein said initiating the timeout process is performed by the timed program execution process.

17. (Currently Amended) The memory medium of claim 1, wherein the rollback state comprises:

a rollback point, comprising an execution point in the program, and

an execution state of the timed program execution process at the rollback point.

18. (Currently Amended) The memory medium of claim 1, wherein the program comprises a subprogram, and wherein said executing the program further comprises:
initiating another timeout process;

configuring another timeout event;

the timed program execution process performing a time-bounded execution of the subprogram, comprising:

f) determining and storing another rollback state for the subprogram;

g) if the another timeout event has not occurred, executing the subprogram, wherein, during said executing the subprogram, if the another timeout event occurs,

h) the another timeout process setting the timed program execution process to the another rollback state, and disabling the another timeout event; and

i) the timed program execution process resuming executing the subprogram based on the another rollback state with another timeout condition;

j) performing a subprogram exit procedure;

disabling the another timeout event; and

terminating the another timeout process.

19. (Currently Amended) The memory medium of claim 18, wherein j) further comprises e).

20. (Currently Amended) The memory medium of claim 1, wherein the ~~medium~~ ~~is further configured~~ program instructions are further executable to perform time-bounded execution of another program substantially in parallel with the time-bounded execution of the program by:

initiating another timeout process;

configuring another timeout event;

the timed program execution process performing a time-bounded execution of the another program, comprising:

f) determining and storing another rollback state for the another program;

g) if the another timeout event has not occurred, executing the another program, wherein, during said executing the another program, if the another timeout event occurs,

h) the another timeout process setting the timed program execution process to the another rollback state, and disabling the another timeout event; and

i) the timed program execution process resuming executing the another program based on the another rollback state with another timeout condition;

j) performing another program exit procedure;
disabling the another timeout event; and
terminating the another timeout process.

21. (Currently Amended) The memory medium of claim 1, wherein said executing the program further comprises:

receiving a disable request from the program to disable the rollback state;
disabling the rollback state in response to said disable request;
receiving an enable request from the program to enable the rollback state;
enabling the rollback state in response to said enable request; and
updating the rollback state for the program.

22. (Currently Amended) The memory medium of claim 1,
wherein the program comprises a machine vision application.

23. (Currently Amended) The memory medium of claim 1,
wherein the program comprises a numerical analysis application.

24. (Currently Amended) The memory medium of claim 1,
wherein the program comprises a text-based program.

25. (Currently Amended) The memory medium of claim 1, wherein the program
comprises a graphical program.

26.-27. (Cancelled)

28. (Currently Amended) A system for performing time-bounded execution of a
program, wherein the system comprises:

a processor; and

a memory coupled to the processor, wherein the memory stores program
instructions executable by the processor to:

initiate a timed program execution process, wherein the timed program
execution process is operable to execute a program in a time-bounded manner;

initiate a timeout process, wherein the timeout process is operable to
preempt the execution process to interrupt execution of the program;

set a timeout event, wherein the timeout event is an event indicating a
timeout condition for the program;

wherein the timed program execution process is operable to perform a time-
bounded execution of the program, wherein, in performing the time-bounded execution,
the timed program execution process is operable to:

a) determine and store a rollback state for the program;

b) if the timeout event has not occurred, execute the program, wherein,
during said executing, if the timeout event occurs,

c) the timeout process is operable to set the timed program
execution process to the rollback state, and disable the timeout event; and

d) the timed program execution process is operable to resume executing the program based on the rollback state with a timeout condition; and

e) perform a program exit procedure;
disable the timeout event;
terminate the timeout process; and
terminate the timed program execution process.

29. (Currently Amended) A system for performing time-bounded execution of a program, comprising:

means for initiating a timed program execution process, wherein the timed program execution process is operable to execute a program in a time-bounded manner;

means for initiating a timeout process, wherein the timeout process is operable to preempt the execution process to interrupt execution of the program;

means setting a timeout event, wherein the timeout event is an event indicating a timeout condition for the program;

means for the timed program execution process performing a time-bounded execution of the program, comprising:

- a) means for determining and storing a rollback state for the program;
 - b) if the timeout event has not occurred, means for executing the program, wherein, during said executing, if the timeout event occurs,
 - c) means for the timeout process setting the timed program execution process to the rollback state, and disabling the timeout event; and
 - d) means for the timed program execution process resuming executing the program based on the rollback state with a timeout condition;
 - e) means for performing a program exit procedure;
- means for disabling the timeout event;

means for terminating the timeout process; and

means for terminating the timed program execution process.

30. (Currently Amended) A computer-accessible memory medium configured to perform time-bounded execution of a program, wherein the memory medium stores program instructions executable by a processor ~~is configured~~ to perform:

setting a timeout condition in a first process;

initiating execution of a program in a second process;

during said execution of the program, the first process determining if the timeout condition has occurred;

if the timeout condition has occurred,

the first process restoring the execution of the program to a pre-defined program state; and

the second process executing the program from the pre-defined state with the timeout condition; and

the program exiting in the second process in response to the timeout condition.